

Stack Ball 3D Documentation

1. Introduction



Stack Breaker is an exciting endless level-base game in which you touch and hold the screen to make player fall down on a colored block, try not to hit the black block or you will die, try to break as many block as you can to enable immortal mode and complete the level. The game is ready to release straight out of the box, and it can also be easily customized to make it even more engaging to your players. Supports for PC/Mac, iOS, Android, etc...!

The game is made with Unity C# and optimized for mobile devices. This template provides you a full-featured, ready-for-release game source code that you can customize and build your own game in no time!

Highlight features:

- ❖ Addictive one-touch gameplay
- ❖ Smooth control, eye-catching graphics
- ❖ 100++ different difficulty levels
- ❖ Very easy to add new levels or edit existing ones
- ❖ Watch ad to revive
- ❖ Admob is integrated (banner, interstitial, rewarded video ad)
- ❖ Optimized for mobile
- ❖ Detailed documentation and commented C# code
- ❖ Free-to-use assets (fonts, sounds, music, models, etc.)
- ❖ Ready to publish out-of-the-box

2. Template set up

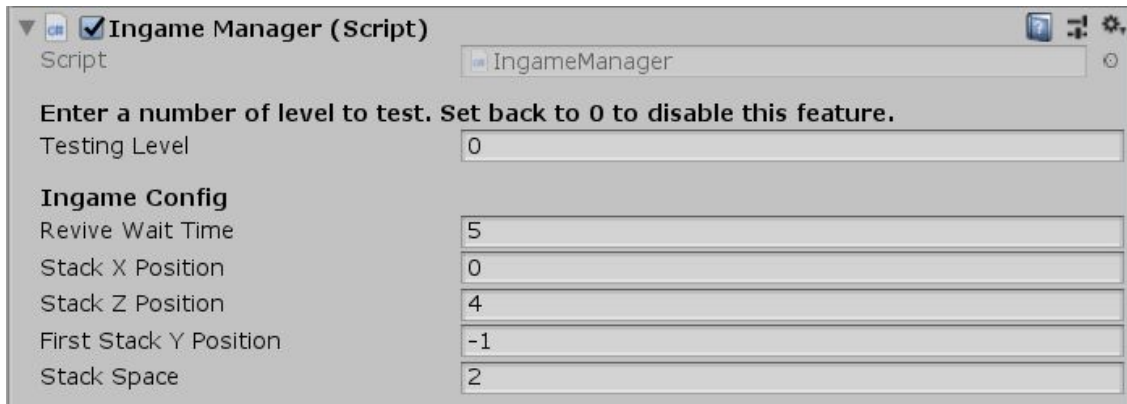
This template was designed for mobile (Android, iOS, Windows Phone...) so after imported the package to unity, you need to switch to Android or iOS.

The template contains 3 scenes: Home, Loading, Ingame under the path Assets/_Stack_Breaker /Scenes. You need to start from Home scene.

3. Gameplay customization

a. Ingame manager

Most of important gameplay parameters can be configured within the IngameManager component which is attached to a game object also named IngameManager in the hierarchy. You can find IngameManager object in InGame scene.



- ❖ **Testing Level:** the level you need to test before building the game. If you set it to 0, the script will not load the testing level.
- ❖ **Revive Wait Time:** the delay time for revive feature last.
- ❖ **Stack X Position:** the position of x axis of all stacks will be created in the game.
- ❖ **Stack X Position:** the position of z axis of all stacks will be created in the game.
- ❖ **First Stack Y Position:** the position of y axis of the first stack will be created in the game. ❖
- Stack Space:** the space between each stack (only effected on y axis).

b. Player Controller

Most of important parameters of player can be configured within the PlayerController component which is attached to a game object named Player in the hierarchy.

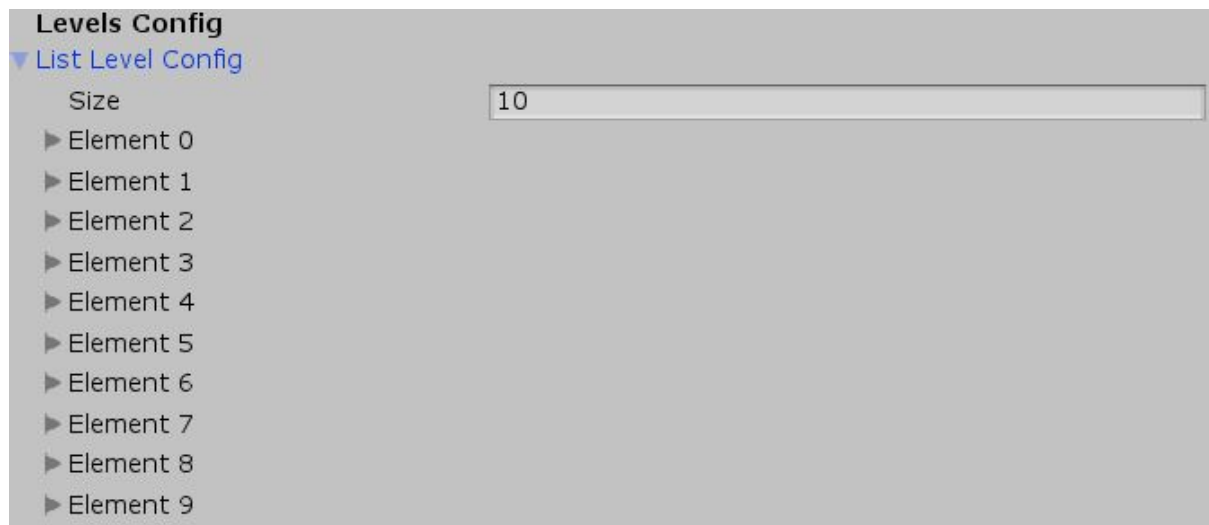


- ❖ **Jump Up Velocity:** the velocity of player when it jumps.
- ❖ **Fall Down Velocity:** the velocity of player when it falls.
- ❖ **Min Scale:** the minimum scale of the player.
- ❖ **Max Scale:** the maximum scale of the player.
- ❖ **Scaling Factor:** the scaling factor of the player.
- ❖ **Time Count To Enable Immortal Mode:** the time player needs to enable immortal mode. This parameter only counts when user holding the screen (player breaking the stack), not bumping on the stack.
- ❖ **Immortal Mode Time:** the time for player performs immortal mode.

4. Level customization

This is a level-base game template and the goal of the game is you have to complete a level and start a new level, try to collect coins and avoid traps and obstacles. All the parameters of the level were show on the inspector for you to adjust and modify. They are stored in

IngameManager script and you can adjust all the levels in the inspector of IngameManager object which you can find it in Ingame scene.



As you can see, there's a list of 10 configuration parameters of levels, these parameters are already designed and ready to use. If you want to add new configs, just resize the list and hit Enter. Now I will show you the parameters of each level.



Level Number Config: this section is about the configuration of level number, each level on this range will use these config parameters bellow to create the level.

- ❖ Min Level: the minimum level of this config.
- ❖ Max Level: the maximum level of this config.

All the level between this range will use these parameters below to create the level.

Colors Config: this section is about the configuration of all the colors needed in this level.

- ❖ Player Color: the color of the player.
- ❖ Deadly Part Color: the color of deadly part in this level.
- ❖ Background Top Color: the color on top of the background.
- ❖ Background Bottom Color: the color on bottom of the background.

Center Pillar Config: this section is about the configuration of the center pillar. These parameters bellow you allow you config the rotation of the pillar.

- ❖ Min Center Pillar Rotating Speed & Max Center Pillar Rotating Speed: the minimum and maximum rotating speed of the center pillar. The actual value will be randomized between these two values.
- ❖ Min Center Pillar Rotating Time & Max Center Pillar Rotating Time: the minimum and maximum rotating time of the center pillar. The actual value will be randomized between these two values.

At first, the center pillar will random a direction to rotate with a randomized speed, it will rotate for a randomized time and stop, then randomized again the time and the speed, then rotate the opposite direction at the start. It will keep rotating like that until player completed the level. It will not stop if the player died.

Stack Configs: this section is about the configuration of all stacks will be created in this level. As you can see there's 3 elements of List Stack Config, so there's will be 3 blocks of stacks will be created. Each block has its own stack number, which define how many stacks will be created and other parameters to config the behavior of that block of stacks. You can add more block by resize the List Stack Config and config the

List Stack Config: this is the list of all configuration parameters of the block of the stacks. Here's the explanation of all parameters.

The screenshot shows a configuration window titled "Stack Configs". It contains two main sections:

- List Stack Config**: This section is expanded, showing a "Size" field with the value "3". Below it, "Element 0" is expanded, showing fields for "Min Stack Number" (20), "Max Stack Number" (25), "Stack Type" (STACK_8_PARTS), "Stack Color" (a teal color swatch), "First Stack Angle" (a slider set to 10), and "Rotation Change Amount" (5).
- List Index Of Deadly Part**: This section is also expanded, showing a "Size" field with the value "2". Below it, "Element 0" has the value "0" and "Element 1" has the value "1".

At the bottom, there are expandable sections for "Element 1" and "Element 2", which are currently collapsed.

- ❖ Min Stack Number & Max Stack Number: the minimum and maximum stack will be created in this level. The actual value will be randomized between these two values.
- ❖ Stack Type: the type of stack you want to create. Currently, the template has 5 stack's type. STACK_6_PARTS, STACK_8_PARTS, STACK_10_PARTS, STACK_12_PARTS and

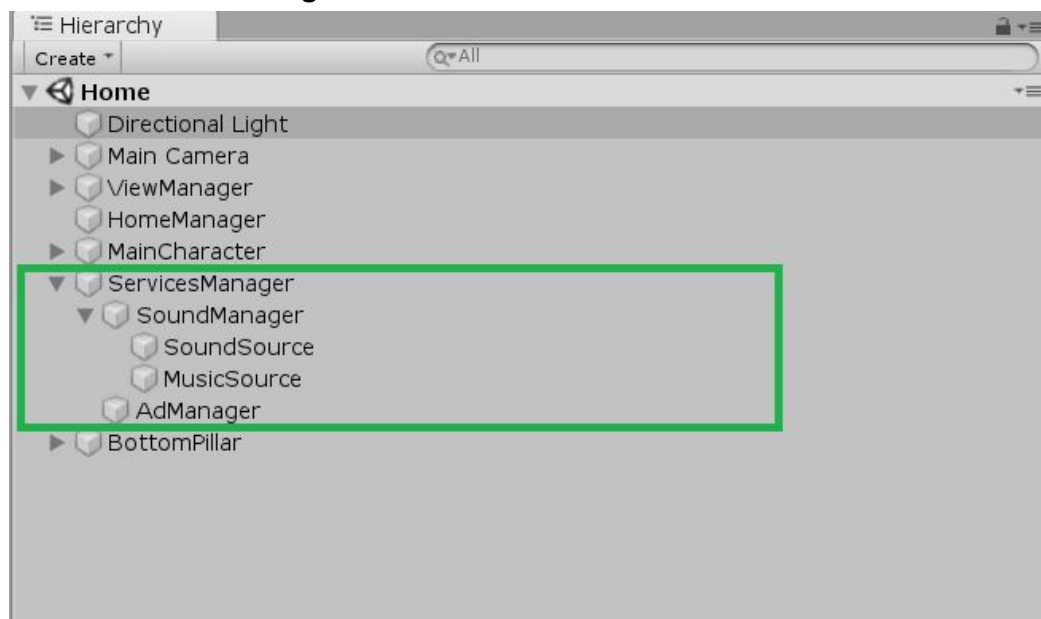
STACK_16_PARTS. The number on the name represent the number of the parts the stack has. Example: STACK_6_PARTS has 6 parts on it.

❖ First Stack Angle: the first angle of first stack of the block.

❖ Rotation Change Amount: the change amount of the angle of each stack. Example: First Stack Angle is 10 and Rotation Change Amount is 5, so the first stack will be created with the angle 10, then the next stack will be created with the angle 15, then the next is angle 20, then angle 25...Of course, you can set the parameter like -5, then the angle of the stacks will be 10, 5, 0, -5, -10...

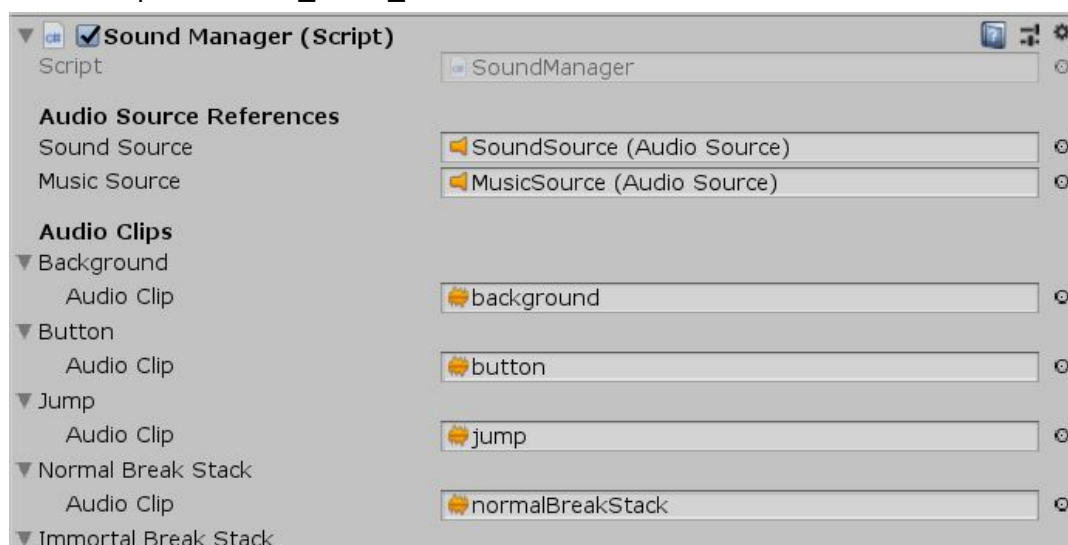
❖ List Index Of Deadly Part: the index of deadly part on each stack, this parameter based on the parameter Stack Type. So, if you set the Stack Type to STACK_6_PARTS, then the stack will have 6 parts on it, so the index of each part will be 0, 1, 2, 3, 4, 5. REMEMBER, THE INDEX WILL START AT 0 SO, YOU CAN'T HAVE THE INDEX 6. As you can see, I set the index to 0 and 1, so the deadly will be the first part and the second part. If I set the index to 1, 3, 5 then the deadly part will be the second part (index 1), the 4th (index 3) and the 6th (index 5). You can set the index to 6 because the index count from 0 to 5.

5. Services configuration



a. Sound Manager

All sounds included in this game are free-to-use in commercial projects and are located under the path Assets/_Stack_Breaker/Audio folder.



b. Ad Manager

Admob is integrated. You need to change the test ad id (default) to your ad id before publishing to the stores.

Ad Manager (Script)	
Script	AdManager
Interstitial Ad Period In Second	75
Admob Controller (Script)	
Script	AdmobController
Banner Id	
Android Banner Id	ca-app-pub-3940256099942544/6300978111
IOS Banner Id	ca-app-pub-3940256099942544/2934735716
Banner Position	Bottom
Interstitial Ad Id	
Android Interstitial Id	ca-app-pub-3940256099942544/1033173712
IOS Interstitial Id	ca-app-pub-3940256099942544/4411468910
Rewarded Base Video Id	
Android Rewarded Base Video Id	ca-app-pub-3940256099942544/5224354917
IOS Rewarded Base Video Id	ca-app-pub-3940256099942544/1712485313